

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra robotiky

Návrh optimalizace elektroniky robotu ARES

Optimizing the Electronics of ARES Robots

Student:

Daniel Huczala

Vedoucí bakalářské práce:

Ing. Ján Babjak, Ph.D.

Ostrava 2016

VŠB - Technická univerzita Ostrava
Fakulta strojní
Katedra robotiky

Zadání bakalářské práce

Student: **Daniel Huczala**
Studijní program: B2341 Strojírenství
Studijní obor: 2301R013 Robotika
Téma: **Návrh optimalizace elektroniky robotu ARES**
Optimizing the Electronics of ARES Robots
Jazyk vypracování: čeština

Zásady pro vypracování:

Analyzujte současný stav elektroniky robotu Ares
Vytvořte úplnou technickou dokumentaci elektroniky robotu
Na základě zjištěných poznatků navrhnete možnosti optimalizace
Zvolenou variantu rozpracujte a zdokumentujte
Práci doložte i v elektronické formě

Seznam doporučené odborné literatury:

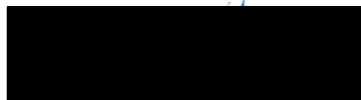
- STUDÉNKA, Marek. 2010. Návrh modulární konstrukce pásového podvozku s proměnnou kinematikou rámu. Ostrava. Diplomová práce. VŠB TU Ostrava. Vedoucí práce Kárník, Ladislav.
- Dokumentace k použitým komponentům
- Dostupná dokumentace k procesorům třídy ARM - STM32

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

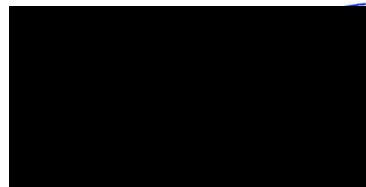
Vedoucí bakalářské práce: **Ing. Ján Babjak, Ph.D.**

Datum zadání: 11.12.2015

Datum odevzdání: 16.05.2016



prof. Dr. Ing. Petr Novák
vedoucí katedry



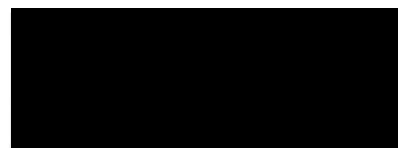
doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty



Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě 16. 5. 2016

A solid black rectangular box used to redact the student's signature.

.....
podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§ 35 odst. 3).
- souhlasím s tím, že bakalářská práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě: 16. 5. 2016



podpis

Jméno a příjmení autora práce: Daniel Huczala

Adresa trvalého pobytu autora práce:



ANOTACE BAKALÁŘSKÉ PRÁCE

HUCZALA, D. *Optimalizace elektroniky robotu ARES: bakalářská práce*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra robotiky, 2016, 54 s. Vedoucí práce: Babjak, J.

Bakalářská práce se zabývá elektronikou robotu ARES. V prvních kapitolách práce jsou popsány jednotlivé součásti, které elektronický systém tvoří. Je vysvětlena aktuální podoba a způsob řízení robotu. Zahrnut je zevrubný výkres elektronického zapojení. V druhé části jsou zmíněny možnosti optimalizace elektroniky. Jedna z navržených variant, v podobě užití platformy Netduino jako hlavního řídicího prvku, je rozpracována. Objasněna je problematika přenosu dat v počítačové síti a přes sériové rozhraní. Součástí práce jsou praktické aplikace pro komunikaci mezi stanicí operátora a robotem, vypracované v programovacím jazyce C#.

ANNOTATION OF BACHELOR THESIS

HUCZALA, D. *Optimizing the Electronics of ARES Robots: Bachelor Thesis*. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Robotics, 2016, 54 p. Thesis head: Babjak, J.

Bachelor thesis is dealing with electronics of ARES robots. In the first chapters are described individual components which form electronic system. The actual form and method of robot driving is explicated. It contains a detailed electronic connection scheme. In second part are mentioned electronic optimizing possibilities. One of suggested options, using Netduino platform as main driving component, is partly done. Issues covering data transmission in computer network and through serial interface are explained. Practical applications for communication between robot and operator's computer are done in C# programming language and included.

Obsah

Seznam použitých značek a symbolů	8
Úvod.....	10
1 Robot Ares	11
2 Prvky robotu Ares	12
2.1 Hlavní počítač	12
2.1.1 Stabilizátor napájení	13
2.2 Subsystem řízení podvozku	14
2.2.1 Motory pohonu	14
2.2.2 Převodovka	16
2.2.3 Kontroléry motorů I-Drive	17
2.3 Subsystem ramene	18
2.3.1 Motory M1, M2, M3	19
2.3.2 Motor M4.....	21
2.3.3 Motor M5 a ovladač efektoru	23
2.3.4 EPOS2	24
2.3.5 EPOS	25
2.4 Zdroj energie.....	26
3 Zapojení prvků robotu Ares	27
3.1 Shrnutí elektronického zapojení a funkčnosti.....	29
4 Možnosti optimalizace	30
4.1 Návrh řešení.....	31
4.1.1 STM32F4 Discovery Kit	31
4.1.2 Netduino plus 2.....	32
4.1.3 Raspberry Pi 3 model B.....	33
4.2 Zvolená varianta	34
5 Komunikace se zařízením Netduino	36
5.1 Přenos dat v počítačové síti	36
5.1.1 Protokol TCP/IP	36
5.1.2 Síťový socket.....	37

5.1.3	Praktické řešení komunikace	37
5.1.4	Ukázka funkčnosti	39
5.1.5	Rychlost komunikace	40
5.2	Sériová komunikace.....	41
5.2.1	Sériová komunikace se subsystémem řízení podvozku.....	41
6	Ověření funkčnosti	44
6.1	Aplikace ClientWF-S.....	44
6.2	Aplikace ServerMF	47
7	Závěr	49
	Seznam použitých pramenů	50
	Seznam příloh	53

Seznam použitých značek a symbolů

Aux	Auxilliary
BIOS	Basic input/output system
BLDC	Bezkartáčový stejnosměrný motor
C#	Programovací jazyk
CAN	Controller area network bus
COM	Communication port
DC	Stejnoseměrný
DDR	Double data rate
EC	Elektronicky komutovaný motor
FSB	Front-side bus
GPIO	General-purpose pnpout/output
HD	High-definition
HDMI	High-definition multimedia interface
IoT	Internet věcí
IP	Internetový protokol
IP XX	Stupeň krytí
LAN	Lokální síť
LVDS	Low-voltage differential signaling
lwIP	Odlehčený internetový protokol
M	Motor
MAC	Media access control
microSD	Typ flashové paměti
OS	Operační systém
PC	Osobní počítač
PCI	Peripheral component interconnect
PCIe	Peripheral component interconnect express

PWM	Pulsně-šířková modulace
QXGA	Quad extended graphics array
RAM	Vyrovňovací paměť počítače
RS-XXX	Standardy pro sériovou komunikaci
SATA	Serial AT attachment
SCTP	Stream control transmission protocol
SSD	Solid-state disk
TCP	Přenosový řídicí protokol
UART	Univerzální asynchronní sériové rozhraní
UDP	Uživatelský datagramový protokol
USB	Univerzální sériová sběrnice
VGA	Video graphics array
Wi-Fi	Bezdrátová lokální síť
WLAN	Bezdrátová lokální síť

Úvod

Vedle konstrukční části je pro každý mobilní robot neméně důležitá elektronická část. Pohony, řídicí prvky, senzory, kabeláž a také způsob, jakým je řízení realizováno, s tím vším je nutno počítat již při návrhu. Jakmile je robot zhotoven a začne být provozně vytěžován, dochází při jeho užívání k nabývání poznatků, co by mohlo vést k vylepšení. Řešeným tématem bakalářské práce je prozkoumání elektronického systému robotu Ares, popsání aktuálního způsobu řízení a vytvoření elektrotechnické dokumentace zapojení dílčích prvků. Následně jsou zváženy některé nedostatky, jež elektronický systém robotu má, a navrženy možnosti jejich optimalizace. Jedna z variant, a to nahrazení řídicího počítače mikrokontrolérem Netduino, je dále rozpracována. Týká se zprovoznění komunikace v počítačové síti pomocí TCP/IP protokolu a sériového odesílání dat přes rozhraní RS-232 na platformě Netduino.

1 Robot Ares

Servisní robot Ares [1] vznikl v roce 2010 po vývoji na katedře robotiky (Fakulta strojní, VŠB – Technická univerzita Ostrava). Jde o univerzální strukturu mobilního robotu s pásovým podvozkem, s jehož pomocí je schopen se pohybovat také po schodech. Podvozková část byla navržena v diplomové práci Marka Studénky [2].



Obr. 1.1 Servisní robot Ares [1]

Robot byl následně doplněn o manipulační nastavbu (rameno), která byla inspirovaná obdobně použitým systémem na robotu Herkules [3]. Možnosti využití jsou velmi široké, robot lze vybavit efektory pro odběr nejrozličnějších vzorků, přístroji pro detekci nebezpečných látek v okolí, dále manipulátory či zásobníky pro odebrané vzorky, jejichž transport umožňuje.

Podvozek:	pásový (12 neodpružených vodících kol)
Řízení:	diferenční řízení (smykem)
Pohon:	každý pás hnaný prostřednictvím jednoho kola
Volitelné rameno:	4 stupně volnosti (rotace) + chapadlo
Motory:	2 DC motory (600 W, 24 V)
Senzory:	kamera vpředu
Řídicí systém:	dálkové bezdrátové řízení všech funkcí z PC
Rozměry:	1253 x 731 x 810 mm

Tab. 1.1 Základní parametry robotu [1]

2 Prvky robotu Ares

2.1 Hlavní počítač

Stěžejním prvkem řídicího PC je základní deska NOVA-GM45A [9], kterou vyrobila firma IEI Integration Corp. Obsahuje socket P pro procesor Intel 45 nm mobile Core 2 Duo. Dále má dva moduly RAM, které jsou v současnosti osazeny 2x4 GB paměti.

Výrobce	IEI Integration Corp.
Výrobní typ	NOVA-GM45A-R10
CPU / MCU	Socket P Intel® 45 nm Core™ 2 Quad/ Duo procesor s 1066/800/667 MHz FSB
FSB	667/800/1066 MHz
Systémový chipset	Intel® GM45+ICH9M
Paměť	2 x 200-pin 800/667 MHz DDR2 (systém max. 4GB)
Grafika	Intel® GMA4500MHD
Obrazový výstup	VGA integrováno v Intel®GM45 s podporou až do QXGA 1 x 18/24-bitový dualní kanál LVDS 1 x HDMI výstup s podporou rozlišení až 1080p
Vstup/výstup	4 x RS232 2 x RS232/422/485 8 x USB 2.0 1 x LPT 4 x SATA II 1 x KB/MS
Digitální výstup	16-bitový digitální vstup/výstup, 8-bitový vstup/výstup
Rozšíření	1 x PCI slot 1 x PC/104+ 1 x PCIe mini card
Super vstup/výstup	ITE IT8718F
Časovač	SW programovatelné 1~255 sec. (systémově resetovatelné)
Napájení	pouze +12 V
Pracovní teplota	0 °C ~ 60 °C (32 ~ 140°F)
Pracovní vlhkost	5 % ~ 95 %, nekondenzující

Tab. 2.1 Parametry základní desky [9]

Ke komunikaci lze využít dva ethernetové porty, až 8 USB výstupů a také rozhraní VGA či případně HDMI. Systém je vybaven SSD diskem firmy Kingston o kapacitě 30 GB, který je připojen přes SATA II rozhraní. Zde je uložen operační systém a aplikace potřebné k obsluze Arese. Pro ovládání a komunikaci s prvky robotu jsou stěžejní porty COM. Na počítači běží OS Microsoft Windows 7, po jeho naběhnutí dochází automaticky ke spuštění řídicích aplikací.

Robot je možné ovládat bezdrátově pomocí herního ovladače od firmy Logitech. Tento gamepad komunikuje pomocí rozhraní Bluetooth, jehož přijímač je připojen k základní desce počítače.

2.1.1 Stabilizátor napájení

Důležitou součástí systému je také stabilizátor napájení pro počítač. Micro DC-DC converter [10] vyrábí firma Vicor Corporation. Přístroj je schopen, kromě redukce napětí, na základě výpočtů a vyhodnocování informací ze zdroje účinně předcházet případnému nebezpečí pro hlavní počítač, především tedy výkyvům napětí.

Modul obsahuje zabezpečovací systém, který monitoruje vstupní napětí, teplotu a své vnitřní operační parametry. V případě, že se jedna ze sledovaných hodnot vychýlí z akceptovatelných mezí, stabilizátor vypne počítač (pošle mu 0 V) a sám sebe uspí. Pracuje poté v režimu, kdy pouze kontroluje vstupní parametry a PC znovu dodá napětí až v případě, že je vše v normálu.

Výrobce	Vicor Corporation
Výrobní označení	V24C12C100BN3
Typ	Standard
Výstupní napětí	12 V
Pracovní teplota	Od -20 °C do +100 °C
Skladovací teplota	Od -40 °C do +125 °C
Výstupní výkon	100 W
Účinnost	Až 0,89
Rozložení pinů	Long ModuMate se zlatým zakončením

Tab. 2.2 Specifikace stabilizátoru Vicor [10]

Výstupní napětí může být regulováno pevnými rezistory, potenciometry anebo převaděči napětí.

2.2 Subsystem řízení podvozku

Subsystem řízení podvozku je ovládán základní deskou, vyrobenou přímo na katedře. Zajišťuje komunikaci na jedné straně s počítačem, tedy hlavním řídicím prvkem a na druhé s regulátory I-Drive (potažmo s motory), ovládá světlomety a blinkry. Rovněž přiděluje pokyny ovladači efektoru, který zajišťuje funkčnost motoru M5 na rameni robotu. Ovladač efektoru byl také vyroben na katedře.

Na obrázku 2.1 na další straně je zpracováno schéma základní desky.

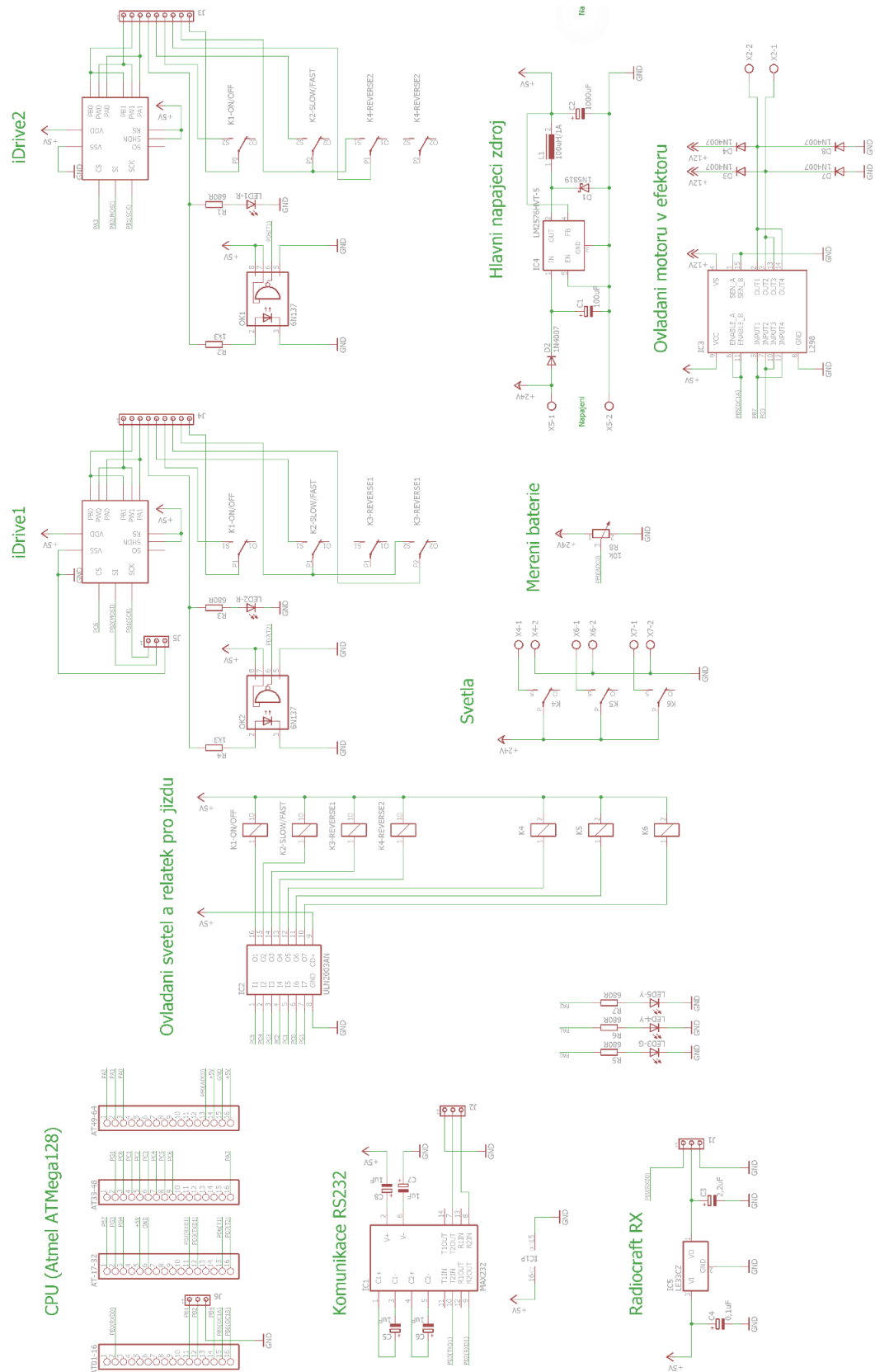
2.2.1 Motory pohonu

Pásový podvozek je poháněn dvěma stejnosměrnými kartáčovými motory P2XR492 [4] (s permanentními magnety) o výkonu 600 W a s přívodem napětí 24 V. Výrobcem těchto motorů je společnost ATAS elektromotory Náchod a.s.

Výrobce	ATAS elektromotory Náchod a.s.
Výrobní označení	P2XR492
Výkon	600 W
Napětí	24 V
Proud	33 A
Otáčky	3600 ot/min
Smysl otáčení	oba
Provoz	S1
Krytí	IP 20
Hmotnost	5 kg

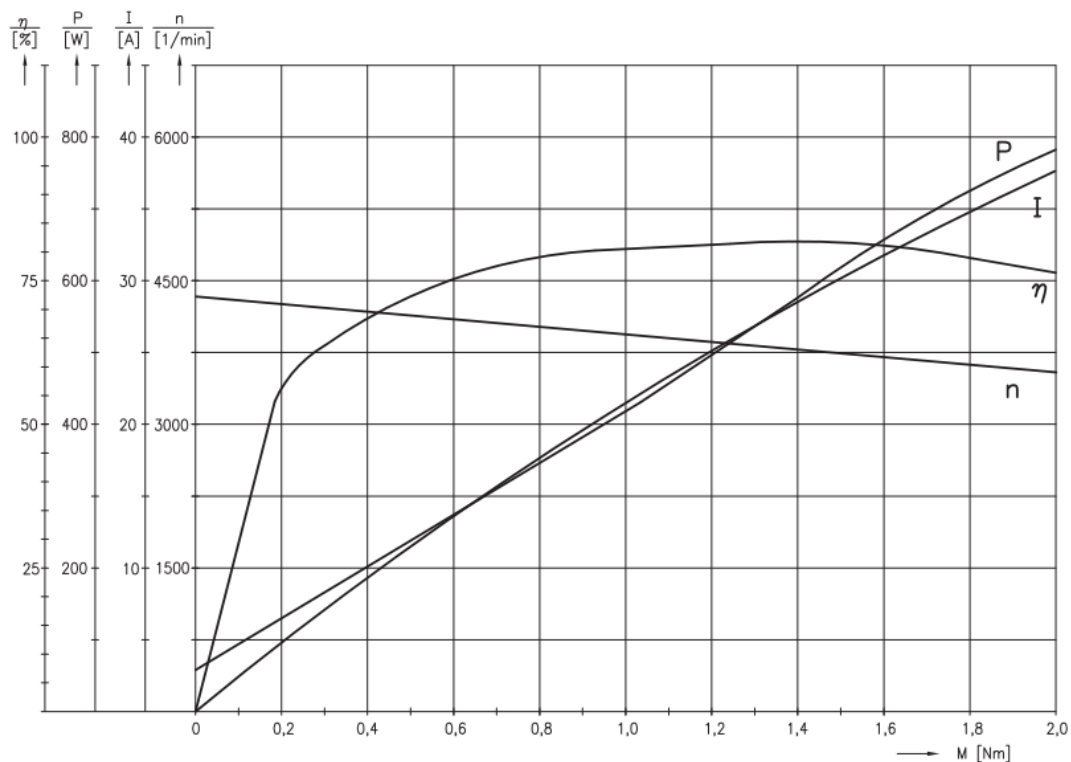
Tab. 2.3 Parametry motoru [4]

Charakteristiky motorů jsou zobrazeny na obrázku 2.2.



Obr. 2.1 Schéma zapojení subsystemu řízení podvozku

P2XR492 24V



Obr. 2.2 Charakteristiky motoru [5]

2.2.2 Převodovka

Výstupní otáčky motoru jsou upraveny šnekovou převodovkou s poměrem $i = 40$. Výrobce převodovky je firma Varvel SpA, jedná se o katalogový typ RS [6].

Výrobce	Varvel Spa
Výrobní označení	SRS05040G525
Sériové číslo	89504
Převodový poměr	40

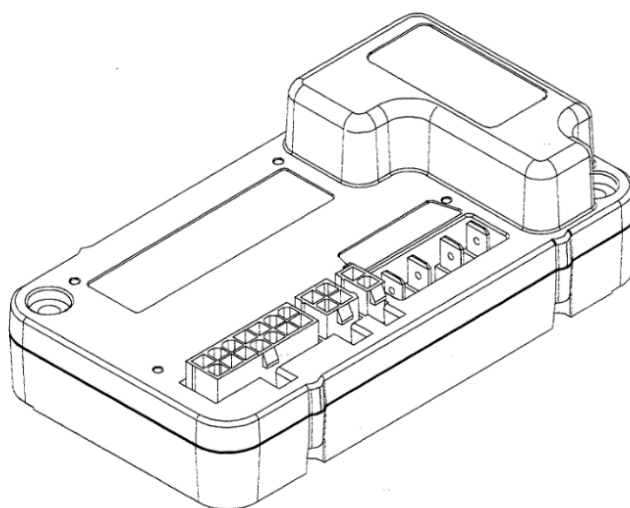
Tab. 2.4 Informace o převodovce [6]

Z označení je možné vyčíst tyto parametry: bez příruby pro rám elektromotoru, katalogová velikost 40, způsob montáže G5, redukční poměr 25.

2.2.3 Kontroléry motorů I-Drive

Každý motor je ovládán prostřednictvím řídicí jednotky motoru I-Drive PCP [7] firmy PG Drives Technology (v roce 2012 odkoupená společností Curtiss-Wright). Použitý typ je schopen pracovat s maximálním výstupním proudem 70 A a poskytuje tři plně programovatelné výstupy pro řízení strojů. Vnitřní elektronika ovladače je zapečetěna proti vniknutí vody. Zařízení je rovněž možné využít pro diagnostiku (kontrola zapojení, měření provozního času, aj.).

Pro nastavení parametrů je k dispozici příslušný software, ovladač se programuje za užití jednoho z konektorů.



Obr. 2.3 I-Drive PCP [8]

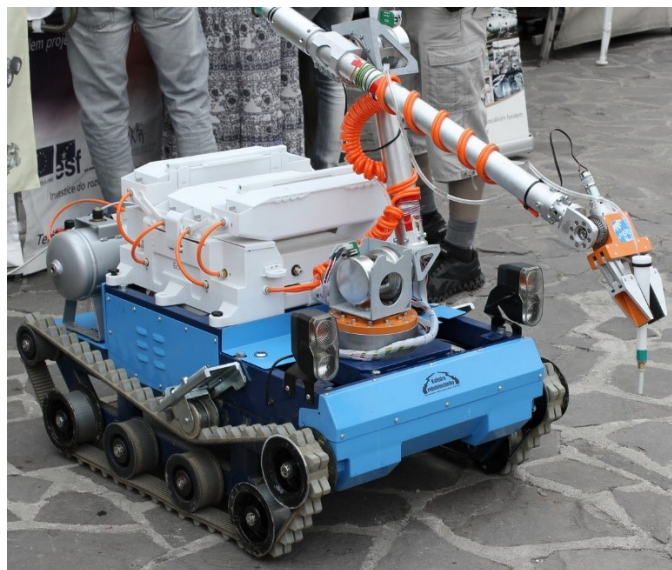
Výrobce	PG Drives Technology (Curtiss-Wright)
Označení	I-Drive PCP
Napájecí napětí	24 Vdc
Pracovní napětí	16 Vdc to 28 Vdc
Špičkové napětí	35 Vdc
Reverzní napětí baterie	40 Vdc
Výstupní proud	45 A / 70 A
PWM frekvence	20 kHz $\pm 1\%$
Aux 1 výstup	Programovatelný mód, 24 V, 1.25 A max.
Aux 2 výstup	Programovatelný mód, 24 V 0.5 A max.
Aux 3 výstup	Programovatelný mód, 12 nebo 24 V, 0.5 A max.
Výstup stavu	Programovatelný 0-12 V, 50 mA zdroj
Vypínací vstup	Programovatelný mód
Konektory zdroje	6.35 mm (0.25") Faston Spade
Konektory Aux 1 výstupní	2-cestný Molex® Mini-Fit, Jr™
Konektor ovládání	14-cestný Molex® Mini-Fit, Jr™
Odolnost proti vlhkosti	Elektronika na IPX5
Krytí	Návaznost na IP X4 (zakrytované vstupy)
Pracovní teplota	Od -25 °C do +50 °C

Tab. 2.5 Specifikace řídicí jednotky motoru I-Drive PCP [7]

2.3 Subsystem ramene

Subsystem ramene je tvořen pěti stejnosměrnými motory s klouby zajišťujícími 4 stupně volnosti. Tři motory jsou bezkartáčové (M1, M2, M3), dva jsou kartáčové (M4, M5). Motor M1 otáčí se základnou ramene. Motor M2 slouží pro pohon prvního kloubu a motor M3 druhého kloubu. Motor M4 zajišťuje ovládání orientačního ústrojí robotu. Motor M5 slouží k řízení efektoru, jako jediný je ovládán přes desku řídicího subsystemu podvozku.

Podmínkou pro užívání bezkartáčových motorů je zařízení, které zajišťuje řízení motoru. U Arese jsou použity řídicí systémy EPOS od firmy Maxon Motor.



Obr. 2.4 Robot Ares s aktuálně instalovaným ramenem [1]

2.3.1 Motory M1, M2, M3

U všech tří stejnosměrných bezkartáčových motorů se jedná o stejný výrobní typ EC 90 flat [14], které vyrábí Maxon motor. Hlavní výhodou motoru jsou rozměry, kdy při průměru 90 mm je výška těla pouhých 30 mm. Motory M1, M2 a M3 jsou řízeny ovladači EPOS2 (kap. 2.5.4). Součástí motoru je také Hallova sonda pro zjištění aktuální polohy natočení motoru a enkodér.

Výrobce	Maxon motor
Výrobní označení	323772
Vnější průměr	90 mm
Napětí	24 V
Výkon	90 W
Rychlost	2590 ot/min
Stálý točivý moment	444 mNm
Stálý proud	6.06 A
Záběrový točivý moment	4940 mNm
Záběrový proud	70 A
Max. účinnost	83 %

Tab. 2.6 Specifikace motoru EC 90 flat [14]

Otáčky motoru jsou redukovány pomocí převodovky Harmonic Drive CSG-20-160-2UH [22] s převodovým poměrem 160:1 a jmenovitým krouticím momentem 52 N·m.

Výrobce	Harmonic Drive AG
Výrobní označení	CSG-20-160-2UH
Vnější průměr	93 mm
Typ ložiska	Se zkříženými válečky
Převodový poměr	160:1
Opakovatelný maximální krouticí moment	120 N·m
Průměrný krouticí moment	64 N·m
Jmenovitý krouticí moment	52 N·m
Kritický krouticí moment	191 N·m
Váha	0,98 kg

Tab. 2.7 Specifikace převodovky HD CSG-20-160-2UH [22]

Pro zjištění polohy či rychlosti je motor vybaven enkodérem MILE [15]. Jedná se o miniaturní zařízení integrované přímo u motoru a nezabírá tak téměř žádné místo navíc. Současně je také odolný vůči elektricko-magnetickým vlivům okolí a zvládne měřit s velkou přesností při vysokých rychlostech.

Výrobce	Maxon motor
Výrobní označení	453231
Počet dílků na otáčku	4096
Max. mechanická rychlost	7250 ot/min
Počet kanálů	2
Napětí	5.0 V ± 10.0 %

Tab. 2.8 Parametry enkodéru MILE [16]



Obr. 2.5 Motor EC 90 flat s enkodérem MILE [15]

2.3.2 Motor M4

Stejnoseměrný kartáčový motor M4 ovládá zápěstí ramene robotu. Výrobce je Maxon motor, katalogový typ RE 35 [17], kartáče jsou grafitové.

Výrobce	Maxon motor
Výrobní označení	323890
Vnější průměr	35 mm
Napětí	24 V
Výkon	90 W
Rychlost	7000 ot/min
Stálý točivý moment	101 mNm
Stálý proud	3.62 A
Záběrový točivý moment	1200 mNm
Záběrový proud	41.1 A
Max. účinnost	86 %

Tab. 2.9 Parametry motoru RE 35 [17]

Redukci otáček zajišťuje planetová převodovka (typ Planetary Gearhead GP 32 C [18]) s poměrem 66:1 a výstupním momentem 6 N·m.

Výrobce	Maxon motor
Výrobní označení	166940
Vnější průměr	32 mm
Verze	Keramická
Převodový poměr	66 : 1
Absolutní převodový poměr	16224/245
Max. průměr hřídele	4 mm
Počet stupňů	3
Max. trvalý krouticí moment	6 Nm
Max. špičkový krouticí moment	7.5 Nm
Max. účinnost	70 %
Max. trvalý převedený výkon	76 W
Max. špičkový převedený výkon	95 W
Max. radiální síla (10 mm od příruby)	140 N
Max. axiální zatížení (dynamické)	120 N

Tab. 2.10 Parametry převodovky GP 32 C [18]

Tento stejnosměrný motor je řízen ovladačem EPOS (kap. 2.5.5), ovšem nejedná se již o druhou generaci, nýbrž první (starší). Ke zjištění úhlu natočení se zde využívá enkodér (typ HEDL 5540 [19]), nikoli však už Hallova sonda. V motoru je zabudována elektromagnetická brzda (typ Brake AB 28 [20]), schopná zamezit samovolnému pohybu kloubu v požadované poloze. Kvůli brzdě je nutné k motoru vést další dva dráty navíc.

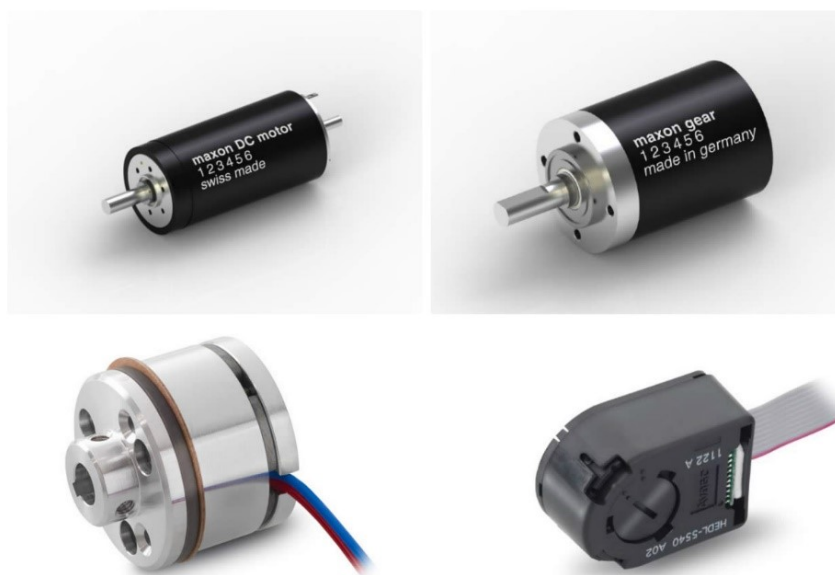
Výrobce	Maxon motor
Výrobní označení	110512
Počet dílků na otáčku	500
Max. mechanická rychlost	12000 ot/min
Napětí	5.0 V \pm 10.0 %

Tab. 2.11 Specifikace enkodéru HEDL 5540 [19]

Výrobce	Maxon motor
Výrobní označení	228387
Napětí	24 V
Brzdicí krouticí moment	400 mNm

Tab. 2.12 Parametry brzdy AB 28 [20]

Popisovaná součást motor, převodovka, enkodér, brzda je dodávána firmou Maxon Motor jako set pod katalogovým označením 341514 [21].



Obr. 2.6 Set motoru [21]

2.3.3 Motor M5 a ovladač efektoru

Tento motor zajišťuje mechanickou funkci efektoru. Jedná se o blíže nespecifikovaný modelářský stejnosměrný motor bez složitější elektroniky pro řízení. Je řízen z příslušného modulu (na schématu 3.1 označen jako „ovladač efektoru“) prostřednictvím pulsně šířkové modulace (PWM).

Zmíněný modul vyrobený na katedře není vybaven žádnými inteligentními prvky (např. rychlost stlačení, aj.) a je zapojen k základní desce řídicího systému podvozku.

2.3.4 EPOS2

EPOS je zařízení sloužící k digitálnímu řízení motorů. Je schopno pracovat jak s kartáčovými DC motory, tak s bezkartáčovými EC motory. Mezi funkce patří například možnost regulace krouticího momentu či možnost krokové změny polohy natočení [13].



Obr. 2.7 EPOS2 [12]

Druhá generace zařízení EPOS2 50/5 [12] je použita pro ovládání motorů M1, M2 a M3. Operuje s daty, které z motoru přivádí Hallova sonda i enkodér, a s příkazy vyvolanými uživatelem. EPOS2 zprostředkovává zjišťování aktuální pozice, rychlosti nebo zrychlení. Obsahuje také senzory pro měření proudů, což jde využít například při případném zkratu (ochranná funkce).

Při současném zapojení Arese je pro komunikaci s hlavním PC využíváno rozhraní RS-232 (PC – EPOS2 #1). Ostatní EPOS jsou zapojeny sériově pomocí sběrnice CAN (EPOS2 #1 – EPOS2 #2 – EPOS2 #3 – EPOS2 #4). Oba konce CAN sběrnice jsou zakončeny terminačními odpory (120 Ω).

Výrobce		Maxon motor	
Výrobní označení		347717	
Váha	240 g	DC/EC motory	5 až 250 W
Pracovní napětí	11 až 50 V	Příkon pro motor	0,9 · (pracovní U)
Max. výstupní I	10 A	Průběžný výstupní I	5 A
Pracovní teplota	-10 až 45 °C	Vlhkost	20 až 80 %
Max. účinnost	0,94		
Operační módy	Proudové, rychlostní, polohové řízení		
Rozhraní	RS-232, USB 2.0, CAN		
Senzory	Enkodér (max. 100 000 rpm), Hallova sonda		

Tab. 2.13 Specifikace ovladače EPOS2 50/5 [12]

2.3.5 EPOS

Motor M4 je řízen systémem EPOS 70/10 [11]. Oproti mladší generaci EPOS2 postrádá USB port či pokročilejší možnosti nastavení a řízení. U motoru M4 je použita z důvodu, že byla dostupná na katedře. EPOS ovládá i elektromagnetickou brzdu, která je součástí motoru.

Výrobce		Maxon motor	
Výrobní označení		300583	
Váha	330 g	DC/EC motory	70 až 250 W
Pracovní napětí	11 až 70 V	Příkon pro motor	0,9 · (pracovní U)
Max. výstupní I	25 A	Průběžný výstupní I	10 A
Pracovní teplota	-10 až 45 °C	Vlhkost	20 až 80 %
Operační módy	Proudové, rychlostní, polohové řízení		
Rozhraní	RS-232, CAN		
Senzory	Enkodér, Hallova sonda		

Tab. 2.14 Specifikace ovladače EPOS 70/10 [11]

2.4 Zdroj energie

Napájení pro počítač, subsystém podvozku i subsystém ramene zajišťují dvě 12 V baterie o kapacitě 40 Ah, zapojené sériově – vytváří tedy zdroj o napětí o hodnotě 24 V. Jedná se o typ baterií LiFePO₄. Mezi nimi se nachází zapojený jistící prvek (tavná pojistka) s hodnotou 40 A.

Napětí je z baterií přivedeno do prvku zvaný Battery management system. Na ten jsou pro odběr elektrické energie napojeny všechny další elektronické součásti. Tento modul (BMS) slouží pro monitorování stavu baterií a chrání je před jejich úplným vybitím (tím, že je odpojí od systémů robota), což by v opačném případě mohlo vést k jejich poškození.

Baterie LiFePO₄ je nutné nabíjet. K tomuto slouží třístupňový přepínač, který odpojí prvky robota od napětí buď do neutrální úrovně, anebo do pozice umožňující pomocí napájecího konektoru baterie nabít.

3 Zapojení prvků robotu Ares

Po analýze současného stavu elektroniky v předchozí kapitole bylo dalším krokem práce vytvoření technické dokumentace elektroniky robotu.

Pro vytvoření podrobného výkresu byl zvolen open-source software KiCad EDA ve verzi 4.0.0 volně dostupné na internetu. Program je zaměřen na tvorbu elektronické dokumentace v podobě schémat zapojení, navrhování tisknutelných obvodů pájivých desek či výpočet a dimenzování elektronických součástek.

Výkres kompletního zapojení robotu Ares je v příloze A – Schéma zapojení elektronických součástí. Kvůli jazykovým omezením softwaru KiCad a především snaze o zachování shodného názvosloví jednotlivých prvků a jejich částí (např. pojmenování vodičů) z původních manuálů výrobců byl výkres vytvořen v anglickém jazyce. Nad razítkem výkresu se nacházejí také obrázky se vzhledem konektorů a číslováním jejich vodičů těchto zařízení:

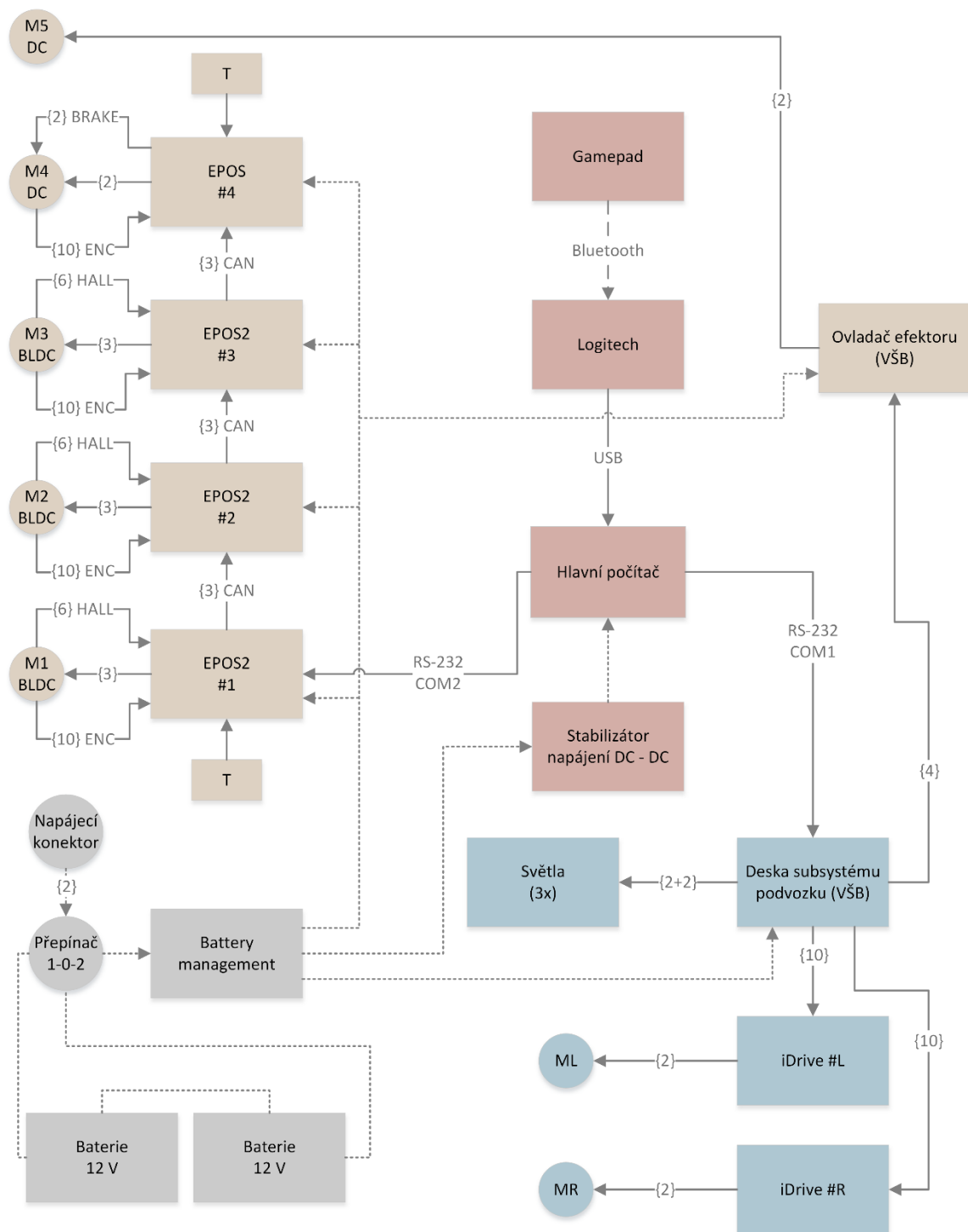
- EPOS 70/10 [11],
- EPOS2 50/5 [12],
- I-Drive PCP [7],
- Maxon EC 90 flat s enkóderem MILE [14].

Pro názornost bylo vytvořeno i zjednodušené schéma zapojení. Výsledek je patrný na obrázku 3.1 další straně.

Legenda k obr. 3.1:

- červená – hlavní počítač se stabilizátorem napětí a řídicími prvky gamepadu
- hnědá – subsystém ramene
- modrá – subsystém podvozku
- šedá – subsystém napájení

Plné čáry značí, že se jedná o výkonový kabel (s počtem vodičů ve složených závorkách). Tečkované čáry vyjadřují přívod napětí z baterií k jednotlivým prvkům.



Obr. 3.1 Znáznornění zapojení elektroniky robotu Ares

3.1 Shrnutí elektronického zapojení a funkčnosti

Ares je řízen operátorem pomocí herního ovladače Logitech přes rozhraní Bluetooth, jehož přijímač je zapojen v hlavním počítači. Hlavní PC vykonává všechny výpočty a pomocí sériové komunikace RS-232 rozdává pokyny dalším subsystémům. Běží na OS Windows 7, po jehož spuštění startují i řídicí aplikace. Data i systém jsou uloženy na SSD disku o kapacitě 30 GB.

Subsystém řízení podvozku je tvořen deskou (napojena na hlavní PC přes COM port), vyrobenou na katedře, motory pásů s jejich ovladači, osvětlením a zasahuje i do řízení efektoru, jehož ovladač (také vyroben na katedře) je na základní desku podvozku připojen. Kontroléry motorů I-Drive nejsou nijak inteligentní, jejich hlavním úkolem je správné zprostředkování diferenčního pohybu pásů. Osvětlení tvoří dvě přední světla a dva blinkry.

Subsystém ramene se skládá z pěti motorů a jejich řídicích jednotek, případně dalších komponentů s nimi souvisejících. Rameno má čtyři stupně volnosti, o správnou a přesnou funkčnost se starají motory Maxon s řídicími jednotkami EPOS, které mezi sebou komunikují na CAN sběrnici. První tři (jdoucí od základny robotu) motory jsou vybaveny Hallovou sondou a enkodérem, pro správné určení pozice za každé situace. Čtvrtý motor postrádá Hallovu sondu, naopak disponuje brzdou. Stlačení dvoučelistového efektoru zajišťuje jednoduchý modelářský stejnosměrný motor s ovladačem vlastní výroby katedry bez složitější inteligence. Hlavní počítač je napojen na první EPOS jednotku přes RS-232.

Subsystém napájení rozvádí napětí z dvou sériově zapojených baterií o celkové kapacitě 24 V do všech prvků elektronického systému robotu Ares. Baterie je možné dobíjet za pomoci třípolohového přepínače. Napájecí kabely se střetávají v jednotce Battery management. Důležitou roli zastává také stabilizátor napětí pro hlavní počítač, který vyrovnává výkyvy odběru energie a plní bezpečnostní funkci. Stejně tak jako pojistka zapojená mezi bateriemi.

Během práce bylo čerpáno z výrobních listů a dokumentů výrobců, od kterých byly jednotlivé elektronické prvky zakoupeny. Odkaz na ně je v seznamu použitých pramenů, některé z nich jsou dostupné v rámci přílohy B na nosiči CD.

4 Možnosti optimalizace

Po zpracování dokumentace, seznámení se s elektronickým systémem robotu a nabytí poznatků v této oblasti, byly zváženy možnosti optimalizace. Ty jsou z elektronického hlediska shledány zejména v těchto oblastech:

- přítomnost „klasického“ počítače s OS Windows 7,
- ovládání pomocí gamepadu skrze rozhraní Bluetooth,
- komunikace ŘS s první EPOSkou pouze přes RS-232.

Řešení užitím hlavní řídicí jednotky v podobě desktopového počítače přímo na robotu by mohlo být upraveno použitím vhodně zvoleného mikrokontroléru. Zamezilo by se nechtěným ztrátám dat, ke kterým náhodně dochází při „tvrdém“ vypnutí systému – řádné vypnutí není zajištěno. Došlo by také ke snížení spotřeby, tj. odběru energie z baterií, a následnému prodloužení výdrže provozu, k úspoře místa a jednodušší montáži řídicí jednotky na robot. Dle typu vybraného mikrokontroléru by mohlo dojít k dalším optimalizacím souvisejícím již s konkrétním případem užití platformy.

Řízení robotu na dálku pomocí herního ovladače Logitech, fungujícím na bezdrátové komunikaci přes Bluetooth rozhraní, je omezující zejména v dosahu. Operátor je povinen se pohybovat ve vzdálenosti několika metrů od robotu. Ares může být vybaven nástavbou pro odběr nebezpečných vzorků a jejich bezpečné uskladnění a transport pro následnou analýzu. Pokud se ovšem operátor musí pohybovat v uvažované blízkosti, může být nasazení robotu např. v kontaminované oblasti komplikované. Řešením by bylo užití rádiové komunikace či připojení a ovládání pomocí Wi-Fi standardu.

U komunikace přes sériové rozhraní RS-232 lze shledat nedostatek v její rychlosti. Jelikož zařízení EPOS mezi sebou komunikují skrze sběrnici CAN, která je schopna dosahovat větších rychlostí, dochází následným předáním informací pomocí RS-232 ke zpomalení celého řídicího subsystému ramene. Pokud by byla sériová komunikace mezi počítačem a první EPOSkou nahrazena komunikací pomocí sběrnice CAN, dospělo by se ke zrychlení celého procesu řízení ramene.

4.1 Návrh řešení

Po dohodě s vedoucím práce bylo rozhodnuto vydat se cestou, která by umožňovala vyřešit aspoň první dva výše zmíněné problémy. Jedním z možných východisek bylo nahrazení stávajícího počítače mikrokontrolérem, který by byl schopen komunikovat přes protokol TCP/IP a v budoucnu by tak šlo využít bezdrátového Wi-Fi přenosu.

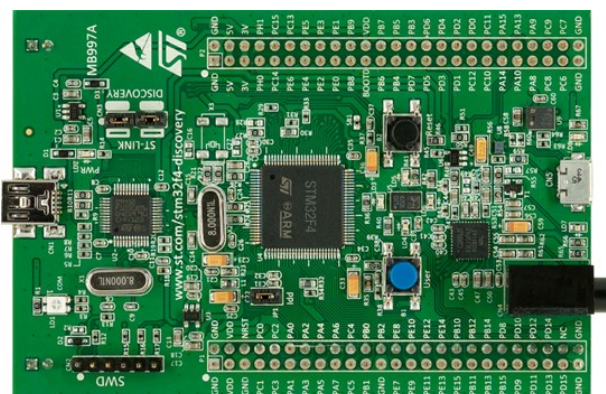
V rámci řešení bylo uvažováno nad třemi různými platformami, které by bylo možné aplikovat. Jejich společným znakem je ne příliš vysoká cena. Jednotlivě a podrobněji jsou rozebrány v následujících podkapitolách. Konkrétně se jedná o tyto platformy:

- STM32F4 Discovery Kit,
- Netduino plus 2,
- Raspberry Pi 3 model B.

4.1.1 STM32F4 Discovery Kit

Mikrokontrolér STM32F407VGT6 [29] ve verzi pod názvem Discovery Kit je poháněn procesorem z rodiny ARM Cortex-M4 s 32-bitovou architekturou a taktem 168 MHz. Je určen pro začátečníky i pokročilé uživatele k vývoji praktických aplikací, jejichž řešení je možné ukládat na flash paměť o velikosti 1 MB. Paměť RAM má velikost 192 KB.

Základní deska nabízí konektivitu v podobě microUSB, obsahuje také programátor ST-LINK/V2. Nastavbami je možné mikrokontrolér obohatit například o Ethernetový port, LCD displej a jiné.



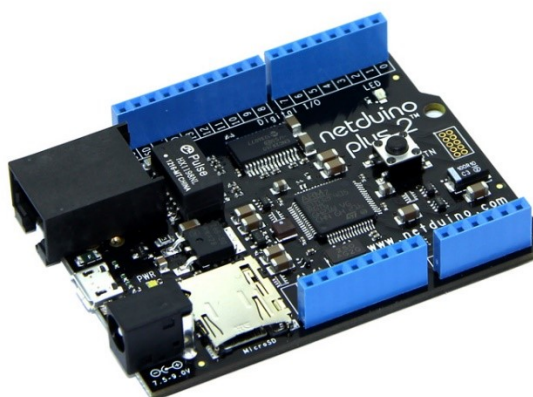
Obr. 4.1 Deska STM32F4 [29]

Mikrokontrolér lze programovat za pomoci mnoha nástrojů (např. Keil MDK-ARM či po úpravách dokonce i v prostředí .NET Micro Framework firmy Microsoft). V uvažovaném zapojení při optimalizaci robotu Ares by měl splňovat dříve stanovené požadavky, užila-li by se nástavba s Ethernetovým portem. Komunikace by se musela programovat. Výhodou platformy je běh aplikace bez operačního systému. Programově se jedná o velmi čisté a efektivní řešení, start trvá pouze jednotky sekund.

4.1.2 Netduino plus 2

Tato open-source platforma je podobná rovněž open-source projektu Arduino a je kompatibilní s jeho nástavbami. Pracuje však na rozdílné technologii a to .NET Micro Frameworku. Stavebním kamenem je 32-bitový procesor o taktu 168 MHz, interní paměť s 384 KB a paměť RAM se 100 KB. Verze plus obsahuje navíc ethernetový port využitelný pro zamýšlenou LAN/WLAN komunikaci.

Jedná se o miniaturní počítač s rozměry 72 x 54 mm, je vybaven [23] digitálními i analogovými vstupy, čtyřmi sériovými porty (UART) či šesti kanály PWM. Data je schopen zapisovat na microSD kartu. Programovat lze nejlépe na desktopovém počítači s OS Windows 7 nebo 8 v prostředí Visual Studio.



Obr. 4.2 Netduino plus 2 [23]

I tento mikrokontrolér nevyžaduje díky předinstalovanému řešení v podobě .NET Micro Framework pro svůj běh instalaci operačního systému. Samotná platforma .NET MF umožňuje a ulehčuje za pomoci předem připravených knihoven tvorbu aplikací pro nejrůznější typy zařízení. Je velmi rozšířená a lze pro ni dohledat velmi rozsáhlé

množství pomocné literatury. I když ovšem Netduino obsahuje ethernetový port, komunikaci by bylo nutné naprogramovat.

4.1.3 Raspberry Pi 3 model B

Platforma Raspberry Pi [24] se od dvou výše zmíněných značně liší. Toto zařízení již může být považováno za regulérní počítač, k jehož obsluze je nejprve nutné nainstalovat operační systém na microSD kartu. Původně byl optimalizován pro linuxové systémy (např. OS Raspbian), nicméně od verze 3 [25] podporuje také Windows 10 ve verzi IoT. K Raspberry lze připojit monitor, klávesnici i myš a pracovat s ním podobně jako s desktopovým počítačem.

Stále se však jedná o malé kompaktní zařízení s rozměry 85 x 56 mm. Pro konektivitu lze využít čtyři USB porty, Ethernetový port, 40 GPIO vstupů, Bluetooth nebo Wi-Fi. Postrádá však analogový vstup. O chod se stará 64-bitový čtyřjádrový procesor ARMv8 o taktu 1,2 GHz. Na tak malé zařízení se jedná o velmi výkonný počítač, který zvládne přehrávat i HD video. Spotřeba je však vyšší než u prvních dvou uvažovaných variant.



Obr. 4.3 Raspberry Pi 3 model B [25]

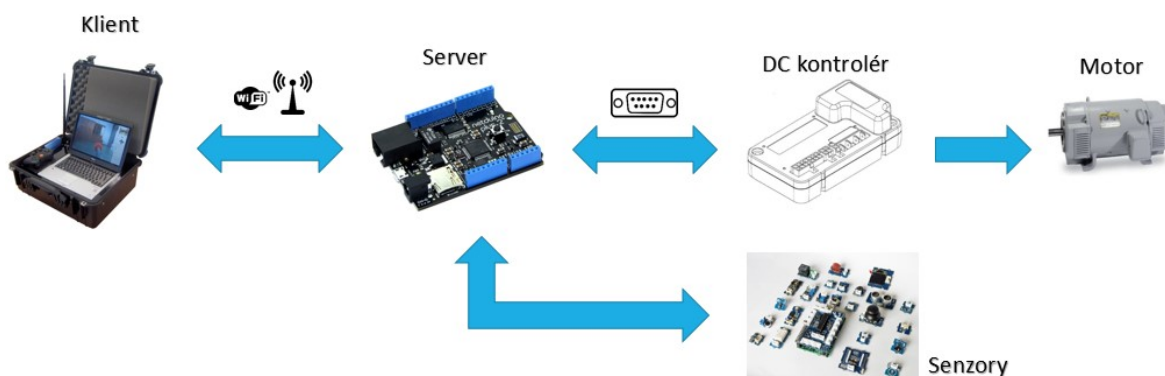
Platforma Raspberry sebou nese řadu výhod v podobě výkonu, konektivity a možnostech použití. Při uvažovaném použití u robotu Ares by ale opět bylo třeba nejprve nastartovat operační systém a až poté požadovanou aplikaci. Řešit by bylo třeba i korektní vypínání zařízení. Naopak odpadá problém s bezdrátovým přenosem dat skrze Wi-Fi, která je již v systému zabudovaná. Nutno ovšem podotknout, že použitý Wi-Fi prvek nelze případně rozšířit o anténu zlepšující signál.

4.2 Zvolená varianta

V rámci řešení prvního a druhého bodu možností optimalizace (nahrazení PC s operačním systémem Windows, nedostatky Bluetooth rozhraní) byla po dohodě s vedoucím práce zvolena varianta s využitím mikrokontroléru Netduino plus 2. Důvody této volby byly především podpora sofistikovaného softwarového řešení v podobě .NET (Micro) Frameworku, dále pak skutečnost, že se na této platformě již několik projektů v rámci katedry řeší a lze tak počítat s případnou využitelností v jiných aplikacích. S tím související roli hrála také dostupnost – katedra Robotiky několik těchto kontrolérů vlastní.

V případě užití nastavby v podobě integrovaného obvodu s CAN konektorem a sběrnici, by platforma Netduino umožnila vyřešení i třetího bodu možností optimalizace, zmíněných v úvodu kapitoly. Tato varianta není součástí této práce, nicméně s ní lze kalkulovat do budoucna.

Princip, na jakém by systém Arese či jakéhokoli jiného robotu v uvažovaném případě fungoval, je znázorněn na obrázku 4.4.



Obr. 4.4 Princip fungování zvolené varianty

Operátor je se svou stanicí umístěn v blízkosti robotu, kterou dovoluje aplikace Wi-Fi přenosu dat. Počítač operátora plní funkci klienta a připojuje se na server, tedy Netduino. Netduino přeposílá data v podobě paketů buď klientovi, nebo pomocí sériové komunikace například kontrolérům stejnosměrných motorů.

Výpočetní výkon zajišťuje počítač operátora. Pokyny lze předávat pomocí dálkového ovladače. Netduino je schopné z kontrolérů motorů také data získávat a přeposílat (například údaje o rychlosti), dovoluje-li to kontrolér. Zvládá také zpracování údajů ze senzorického subsystému.

Pro zajištění funkčnosti navrhovaného principu ovládání pomocí platformy Netduino, je potřeba vypracovat:

- komunikaci pomocí TCP/IP protokolu s počítačem operátora,
- sériovou komunikaci s kontroléry robotu Ares.

Z těchto dvou záležitostí je komplikovanější zprovoznění síťové komunikace. Bylo jí proto v rámci této práce věnováno více prostoru.

5 Komunikace se zařízením Netduino

Netduino plus 2 obsahuje Ethernetový konektor, který lze využít ke komunikaci s dalšími zařízeními. Rychlost přenosu je 10 Mbit/s [23]. Aplikován je protokolový zásobník (z anglického „protocol stack“) lwIP [27], což je odlehčená implementace protokolu TCP/IP vhodná pro miniaturní zařízení s omezenou kapacitou paměti. Přenos dat pomocí této technologie je užitečný zejména z důvodu softwarového řešení, kdy je zajištěn přenos dat ve správné podobě (bez chyb) již samotnými protokoly.

5.1 Přenos dat v počítačové síti

Počítačová síť je skupina připojených zařízení jako jsou počítače a tiskárny, které spolu komunikují. Protokol je skupina pravidel, která v počítačové síti řídí datovou komunikaci. Definuje předmět, způsob a dobu komunikace [26].

5.1.1 Protokol TCP/IP

TCP/IP – z anglického „Transmission Control Protocol/Internetworking Protocol“ je sada protokolů pro komunikaci v počítačové síti. Skládá se z pěti vrstev – aplikační, transportní, síťové, linkové (z anglického „datalink“) a fyzické. Poslední dvě jmenované nemají v rámci TCP/IP definovaná žádná další specifika a podporují všechny standardy pro ně určené, např. Local Area Network (LAN). Pracují s fyzickou neboli MAC (Media Access Control) adresou zařízení [26].

Síťová vrstva pracuje s protokolem IP. Je zodpovědná za doručení paketu mezi zařízeními a pracuje s jejich logickými (IP) adresami [26].

Transportní vrstva zajišťuje doručení zprávy mezi procesy (běžícími programy), pracuje s protokoly TCP, UDP (User Datagram Protocol) či případně SCTP (Stream Control Transmission Protocol) a operuje s adresou (číslem) portu [26].

5.1.2 Síťový socket

Socket je objekt reprezentující nízkoúrovňový přístupový bod k IP zásobníku. Může být otevřený, zavřený anebo se může nacházet v jednom z mnoha přechodových stavů. Socket může během spojení odesílat a přijímat datové pakety [28].

Adresa socketu je kombinace IP adresy a čísla portu. Definuje unikátní označení pro server nebo klienta [26].

5.1.3 Praktické řešení komunikace

Pro požadovanou funkčnost byly vytvořeny dvě aplikace v programovacím jazyce C#. Jedna z nich je spuštěna na Netduinu a funguje jako webový server, druhá běží na počítači operátora a chová se jako klient. Pro Netduino není dostupné žádné předem hotové softwarové řešení zprostředkovávající komunikaci skrze TCP/IP protokol. Využít lze ovšem připravené jmenné prostředí System.Net.Sockets, obsahující třídy k tvorbě aplikací pracujících s rozhraním Windows Sockets.

Na zjednodušených diagramech aplikací ServerMF a ClientWF-S na obrázku 5.1 na další straně lze vidět průběh komunikace.

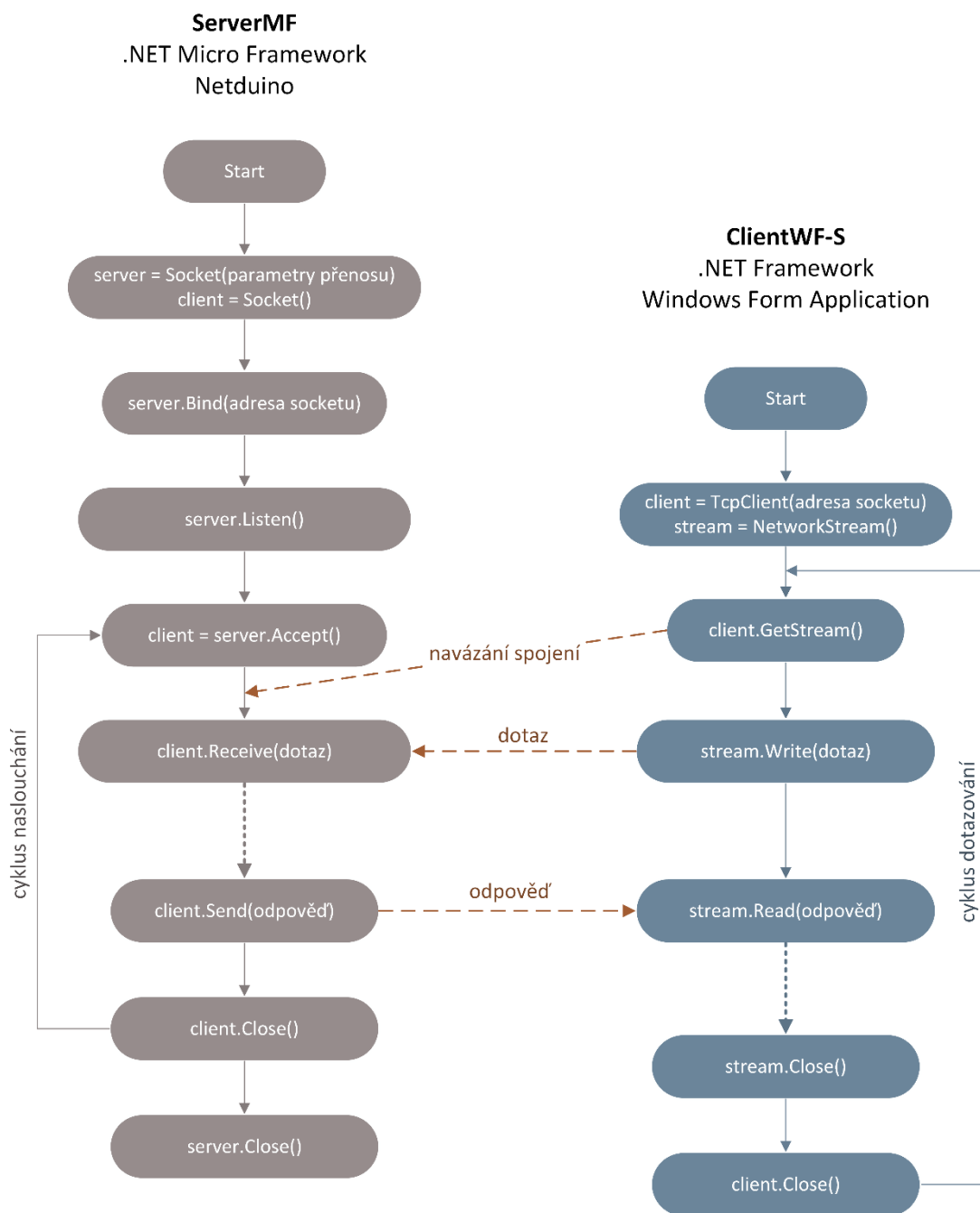
ServerMF běží na Netduinu na platformě .NET Micro Framework. Po startu aplikace jsou definovány objekty server a client ze třídy Socket. Užité parametry přenosu jsou:

- AddressFamily – InternetNetwork
- SocketType – Stream
- ProtocolType – Tcp

Dále následují kroky „server.Bind()“ specifikující adresu socketu a „server.Listen()“ spouštějící cyklus naslouchání. V programu ClientWF-S jsou nejprve definovány objekty client ze třídy TcpClient a stream ze třídy NetworkStream. Obě třídy pocházejí také ze jmenného prostředí System.Net.Sockets.

Po nastavení parametrů komunikace začíná server naslouchat a čekat na připojení klienta (krok „server.Accept()“). Spojení je navázáno klientem v kroku „client.GetStream()“,

a pokud vše proběhlo úspěšně, dochází ihned k odeslání dotazu klientem na server. Zde je tento požadavek zpracován jinými částmi kódu (tečkované šipky; záleží na aplikaci) a dochází k odeslání odpovědi serveru klientovi. Po obdržení zprávy je spojení ukončeno, program ServerMF se ihned v rámci cyklu naslouchání vrací do pozice, kdy je schopen přijmout dalšího klienta. ProgramWF-S po zpracování odpovědi ukončí část dotazovacího cyklu a vrací se do části kódu, kde je specifikováno, např. kdy má dojít k dalšímu spojení.



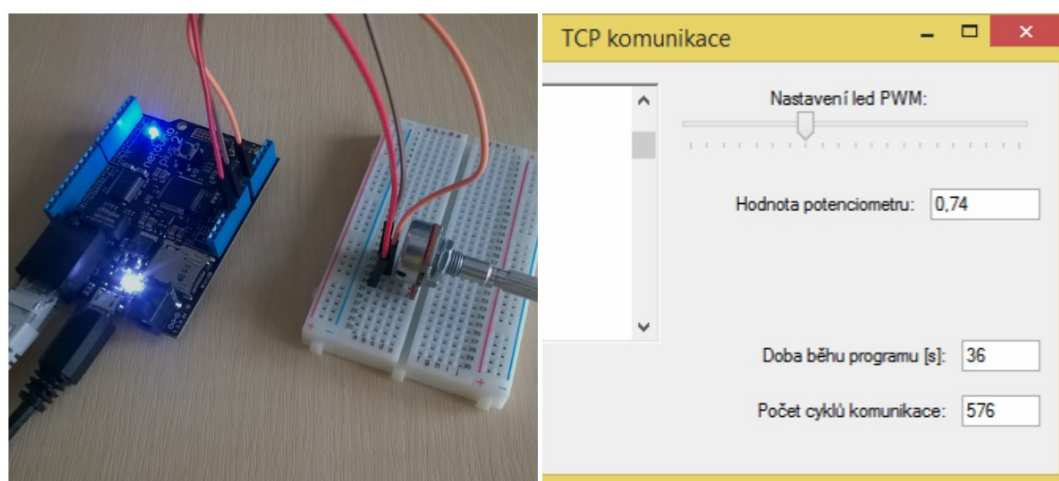
Obr. 5.1 Diagram s programy a aplikovanou variantou komunikace

Výše popsaná varianta přenosu dat byla shledána jako optimální. V průběhu tvorby programů bylo experimentováno i s možností trvalého připojení, kdy by mohl server i klient kdykoli odeslat zprávu nezávisle na sobě a druhá strana by ji vzápětí přijala (v angličtině nazýváno „handshake“ komunikací). Vyskytovaly se ale problémy s nestabilitou aplikací serveru i klienta vylučující možnost praktického nasazení.

U programu ClientWF-S si lze všimnout rozdílnosti při počáteční definici objektů a také ve způsobu přijímání a odesílání dat. Je to způsobeno užitím platformy .NET Framework, tedy odlišné verze, než je verze Micro běžící na Netduinu. .NET Micro Framework je odlehčená varianta pro méně výkonná zařízení, která neobsahuje všechny předdefinované třídy, ale pouze ty základní. Například třídy TcpClient a NetworkStream vycházejí ze třídy Socket ze stejného jmenného prostoru a jsou určeny pro konkrétní použití. V .NET Micro Frameworku ale nejsou obsaženy. To bylo důvodem použití obecné třídy Socket.

5.1.4 Ukázka funkčnosti

Dle požadavků, kdy má být Netduino schopno data správně odesílat i přijímat, byla vytvořena demonstrační úloha. Na mikrokontrolér se připojil potenciometr, jehož data byla analogově zpracována a odesílána do aplikace klienta. Klient měl možnost pomocí posuvníku a PWM modulace regulovat intenzitu svitu led diody na Netduinu.



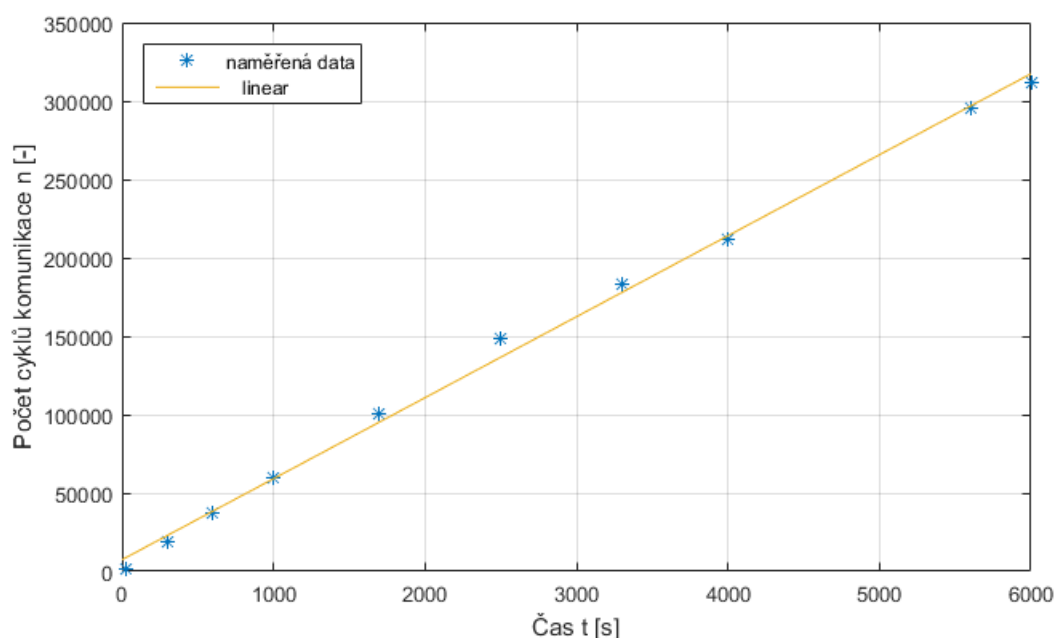
Obr. 5.2 Zapojení a ukázka aplikace klienta demonstrační úlohy

Zdrojové kódy demonstrační úlohy jsou k nahlédnutí v příloze C. Obsahují aplikace pro Netduino i operátora.

5.1.5 Rychlost komunikace

Čas odesílání je specifikován nastavením časovače. Pro posouzení způsobilosti použití aplikace bylo třeba také ověřit rychlost, s jakou je schopno Netduino s počítačem operátora v počítačové síti komunikovat.

Měření bylo provedeno způsobem, kdy počet proběhlých cyklů komunikace korespondoval s časovým úsekem v sekundách od spuštění programu TCP komunikace (demonstrační úloha). Výsledkem byla lineární závislost počtu cyklů komunikace na době běhu programu (obr. 5.3).



Obr. 5.3 Graf závislosti počtu cyklů komunikace na době běhu programu

Z tabulky 5.1 s naměřenými hodnotami je ovšem patrné, že se rychlost komunikace s dlouhodobým během aplikace snižuje – její frekvence, představující počet cyklů za sekundu, klesá. Byla proto stanovena hodnota doporučené frekvence 30 Hz, kdy jeden cyklus dotaz – odpověď trvá přibližně 0,033 s a pro zvolené řešení je dostačující.

Měření	1	2	3	4
Doba běhu programu [s]	30	300	600	1000
Počet cyklů komunikace [-]	1855	18465	37284	59241
Frekvence odesílání [Hz]	61,83	61,55	62,14	59,24

Měření	5	6	7	8	9	10
t [s]	1700	2500	3300	4000	5600	6000
n [-]	100741	148610	182877	211851	295000	311524
f [Hz]	59,26	59,44	55,42	52,96	52,68	51,92

Tab. 5.1 Naměřené hodnoty rychlosti komunikace

5.2 Sériová komunikace

Sériová komunikace je typ přenosu dat, uskutečněný postupně bit po bitu, tedy v sekvencích [31].

Aby byla sériová komunikace mezi zařízeními od různých výrobců kompatibilní, byly vyvinuty různé normy. Většina zařízení používá normu RS-232-C, kterou publikovala v roce 1969 Electronic Industries Association [30]. Tato norma specifikuje signálové napětí, signálový čas, signální funkce, protokoly pro informační výměnu a fyzické konektory [31].

Netduino plus 2 obsahuje [23] v rámci svých digitálních vstupů a výstupů asynchronní sériové rozhraní UART (z angličtiny „Universal Asynchronous Receiver and Transmitter“), umožňující sériovou komunikaci.

5.2.1 Sériová komunikace se subsystémem řízení podvozku

Pro dokázání funkčnosti ovládání robotu Ares mikrokontrolérem Netduino byla vybrána varianta se zapojením k základní desce subsystému podvozku. V současném stavu probíhá komunikace pomocí COM portů, konkrétně u Netduina na portu COM1.

Parametry komunikace jsou tyto:

- rychlost přenosu 9600 baudů,
- bez parity,
- 8 datových bitů,
- jeden stop bit.

Pro požadované chování robota při komunikaci po sériové lince je důležité odesílání paketu dat ve správném tvaru. Deska subsystému řízení podvozku je připravena pro příjem dat dle této šablony:

>XXYYZZQ\r

Význam řetězce znaků je následující:

- > – počáteční znak paketu,
- XX – rychlost jednoho pásu,
- YY – rychlost druhého pásu,
- ZZ – řídicí prvek,
- Q – ovládání efektoru,
- \r – odřádkování, značí konec paketu.

Části XX, YY a ZZ jsou čísla ze šestnáctkové (hexadecimální) soustavy. Mohou nabývat hodnot 00 až FF, tedy 0 až 255. Q nabývá hodnot od 0 do F, tedy 0 až 15, čímž se reguluje síla stlačení a směr pohybu čelistí efektoru.

Je nutné se pozastavit u řídicího prvku řetězce ZZ a objasnit jeho význam. Jeho pomocí lze ovládat zapnutí světel a blinkrů, povolit provoz motorů a měnit jejich směr otáčení.

Číslo bitu	7	6	5	4	3	2	1	0
Váha	128	64	32	16	8	4	2	1
Nastavení (příklad níže)	0	0	0	1	0	1	1	1

Tab. 5.2 Váhy jednotlivých bitů

Význam bitů v tabulce 5.2:

- 0 – zapnutí/vypnutí světel,
- 1 – zapnutí/vypnutí jednoho blinkru,
- 2 – zapnutí/vypnutí druhého blinkru,
- 3 – (žádný),

- 4 – zapnutí/vypnutí motorů,
- 5 – směr otáčení jednoho motoru,
- 6 – směr otáčení druhého motoru,
- 7 – (žádný).

Příklad užití: je-li požadováno spuštění motorů v přímém směru, nastaví se bit číslo 4 na kladnou hodnotu. Získá tak váhu 16, která je v hexadecimální soustavě rovna číslu 10. Dále chceme Zapnout světla a rovněž obě směrovky. Bity jim náležící jsou nastaveny na kladnou hodnotu a získávají společnou váhu $1 + 2 + 4 = 7$. Sedmička je v hexadecimální soustavě neměnná. Výsledný tvar části ZZ je poté roven hexadecimálnímu součtu čísel $10 + 7 = 17$.

6 Ověření funkčnosti

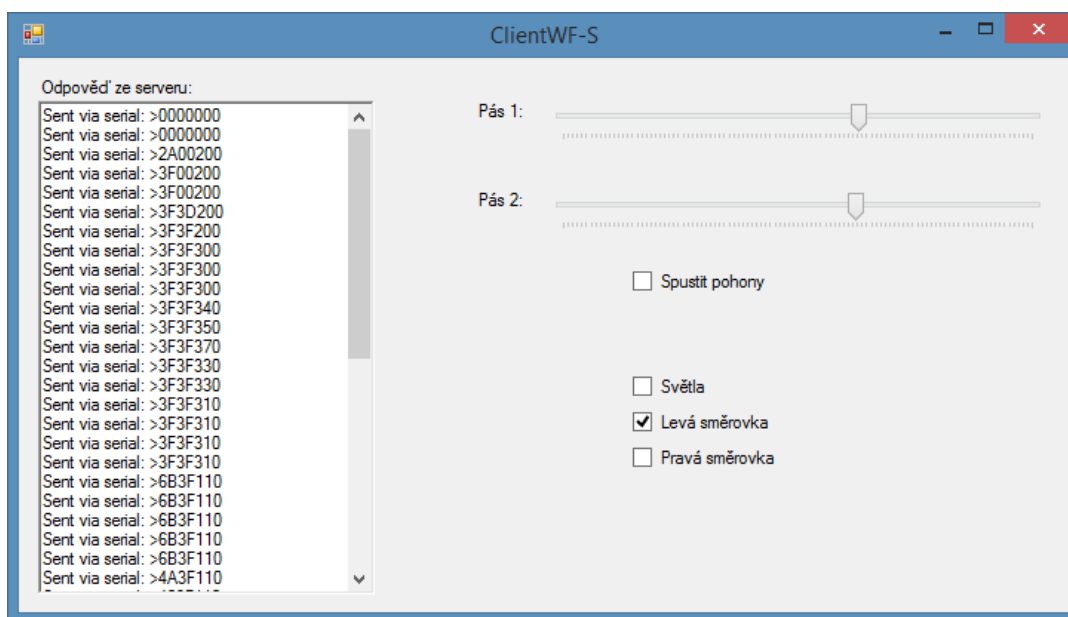
Posledním krokem práce bylo testování správného chování robotu Ares po připojení mikrokontroléru Netduino k subsystému řízení povozku. Testy proběhly úspěšně.

Na počítači operátora, který je připojen ke školní síti, je spuštěna řídicí aplikace. Ta odesílala pakety dat skrze TCP/IP protokol na vzdáleně připojené Netduino (rovněž ve školní síti). Netduino paket zpracovává a přes sériové rozhraní jej posílá na připojenou desku subsystému řízení podvozku. Ares reaguje dle očekávání. Netduino o úspěšném odeslání informuje operátora v odpovědi formou zprávy. Při testu nečekaného přerušení spojení byl díky ošetření na straně subsystému podvozku provoz Arese zastaven.

Vytvořené programy pracují dle principů popsanych v předchozích kapitolách. Jejich zdrojové kódy jsou obsaženy v příloze D.

6.1 Aplikace ClientWF-S

Program ClientWF-S fungující na PC operátora se připojuje jako klient na vzdálený server, jenž běží na Netduinu. Odesílání dat je vykonáváno v závislosti na rychlosti komunikace, která dosahuje času přenosu až 20 ms na jednu zprávu.



Obr. 6.1 Klientská aplikace

Páteří programu jsou čtyři třídy, z nichž dvě jsou systémové (Program.cs, Form1.cs) a dvě doplněné (Connecting.cs, Reading.cs). Hlavní třída Program.cs spouští třídu formuláře Form1.cs, kde dochází k inicializaci základních komponentů, zadání adresy socketu serveru (IP a port), startu časovače, pomocí kterého lze upravovat frekvenci odesílání zpráv, a definici událostí, při kterých jsou vykonávány příslušné operace. Je zde také část kódu, která slouží k tvorbě paketu pro subsystém podvozku při každém tiky časovače.

```
private void myTimer_Tick(object sender, EventArgs e)
{
    byte control = 0;
    string tx_message;

    // Smer jednoho pasu
    if (trackBar_pas1.Value < 0)
        control |= 32;
    // Smer druhého pasu
    if (trackBar_pas2.Value < 0)
        control |= 64;

    // Ovladani motoru enable
    if (checkBox_enable.Checked)
        control |= 16;

    // Svetla
    if (checkBox_svetla.Checked)
        control |= 1;
    if (checkBox_blinkerL.Checked)
        control |= 2;
    if (checkBox_blinkerR.Checked)
        control |= 4;

    tx_message = ">";
    tx_message += Math.Abs(trackBar_pas1.Value).ToString("X2");
    tx_message += Math.Abs(trackBar_pas2.Value).ToString("X2");
    tx_message += control.ToString("X2");
    tx_message += "0\r";

    connecting.SendData(tx_message);
}
```

Ukázka kódu 6.1 Vznik paketu pro subsystém podvozku

Data jsou odeslána pomocí třídy Connecting.cs. Ta slouží k navázání spojení, předání a příjmu informací. Ještě před odesláním ji lze obohatit o různé části upravující finální paket,

který bude poslán do počítačové sítě. Při současném provozu to zatím nemá smysl. V ukázce kódu 6.2 jsou k nahlédnutí nejdůležitější části třídy zajišťující správnou funkčnost. Jedná se ale pouze o několik málo řádků s vynecháním podstatných oddílů.

Navázání spojení	<pre>private void Connect() { try { client = new TcpClient(_server, _port); stream = client.GetStream(); } : }</pre>
Odeslání dat	<pre>public void SendData(String dataS) { : // Prevod na bytes byte[] msg = System.Text.Encoding.ASCII.GetBytes(dataS); // Odesli zpravu stream.Write(msg, 0, msg.Length); : }</pre>
Příjem dat	<pre>if (stream.DataAvailable) { // Nacteni dat ze streamu byte[] data = new byte[client.ReceiveBufferSize]; Int32 bytes = stream.Read(data, 0, data.Length); // Prevedeni odpovedi na textovy retezec responseData = System.Text.Encoding.ASCII.GetString(data, 0, bytes); // Predani dat ke zpracovani reading.Read(responseData); : }</pre>
Ukončení spojení	<pre>stream.Close(); client.Close();</pre>

Ukázka kódu 6.2 Důležité části třídy Connecting.cs

Poslední třída Reading.cs je v současné podobě velmi jednoduchá. Slouží pouze ke zpracování a zobrazení zprávy v textovém okně aplikace ClientWF-S. Je nicméně připravena pro případné rozšíření.

Třídy využívají během své práce jmenné prostory (knihovny) zapsané v tabulce 6.1.

System; System.Collections.Generic; System.ComponentModel; System.Data; System.Drawing;	System.Linq; System.Text; System.Threading.Tasks; System.Windows.Forms;	System.IO; System.Net; System.Net.Sockets; System.Threading;
---	--	---

Tab. 6.1 Jmenné prostory programu ClientWF-S

6.2 Aplikace ServerMF

Na Netduinu se běžící program skládá ze tří tříd – základní Program.cs, dále pak Connecting.cs a Reading.cs.

Třída Program.cs má za úkol pouze spustit aplikaci a předat po připojení do počítačové sítě adresu socketu třídě Connecting.cs. Ta se stará o správnou inicializaci serveru, příjmu dat, jejich předání třídě Reading.cs a finálním odesláním odpovědi operátorovi.

Inicializace serveru	<pre>// Nastavení IP adresy a nastavení parametru serveru IPAddress localAddr = IPAddress.Parse(_serverIP); server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp); server.Bind(new IPEndPoint(localAddr, _port)); // Pocatek naslouchani server.Listen(10); // Cekani na klienta client = server.Accept();</pre>
Čtení dat	<pre>// Kontrola, zda jsou v siti data pro cteni int bytesReceived = client.Available; if (bytesReceived > 0) { // Nacteni dat byte[] data = new byte[bytesReceived]; int bytes = client.Receive(data, bytesReceived, SocketFlags.None); responseData = new string(Encoding.UTF8.GetChars(data)); // Predani dat v podobe textového reztezce ke zpracovani reading = new Reading(responseData); }</pre>

Ukázka kódu 6.3 I. Část kódu třídy Connecting.cs I

Odeslání odpovědi a ukončení spojení	<pre>String dataS = "Odpoved"; client.Send(Encoding.UTF8.GetBytes(dataS), dataS.Length, SocketFlags.None); client.Close(); : server.Close();</pre>
--------------------------------------	---

Ukázka kódu 6.4 Část kódu třídy Connecting.cs II

Třída Reading.cs vykonává operace mezi příjmem a odesláním odpovědi přes počítačovou síť. Mimo jiné zprostředkovává přenos dat subsystému řízení podvozku, realizovaném sériovou komunikací. Řešení přenosu je zobrazen v ukázce kódu 6.5. Program lze rozšířit o čtení dat ze sériového portu, avšak při testování provozu se subsystémem povozku jej nebylo nutné aplikovat.

<pre>// Overeni otevreni portu if (!com.IsOpen) { com.Open(); } // Prevod dat na bytes a odeslani dat byte[] sendSerial = Encoding.UTF8.GetBytes(_responseData); com.Write(sendSerial, 0, sendSerial.Length);</pre>
--

Ukázka kódu 6.5 Odesílání dat přes sériový port

Užité jmenné prostory aplikace ServerMF vychází z předinstalovaného .Net Micro Frameworku. Jsou ale také obsaženy některé navíc, jejich seznam je v tabulce 6.2.

System; System.Net; System.Net.Sockets; System.Threading;	Microsoft.SPOT; Microsoft.SPOT.Hardware; SecretLabs.NETMF.Hardware; SecretLabs.NETMF.Hardware.Netduino;	System.Text; System.IO; System.IO.Ports;
--	--	--

Tab. 6.2 Jmenné prostory programu ServerMF

7 Závěr

V úvodních kapitolách bakalářské práce byl analyzován současný stav elektroniky robotu Ares. Tyto kapitoly obsahují stručný popis jednotlivých částí doložený případnými tabulkami s vlastnostmi. Výrobní listy a dokumenty, z nichž bylo čerpáno, jsou přiloženy na CD.

Zpracován a přiložen je dle zadání také podrobný výkres zapojení jednotlivých elektronických prvků.

V další části práce byly zmíněny některé nedostatky elektronického subsystému. Jednalo se o přítomnost „klasického“ počítače s OS Windows 7, ovládání pomocí gamepadu skrze rozhraní Bluetooth a komunikaci řídicího subsystému s první EPOSkou pomalejším rozhraním RS-232. Byly navrženy možnosti optimalizace v těchto oblastech a přistoupilo se k řešení prvních dvou jmenovaných problémů s tím, že komunikace s první EPOSkou bude v rámci zvolené varianty možné řešit do budoucna.

Zvolená varianta k rozpracování sestávala v nahrazení hlavního řídicího počítače mikrokontrolérem Netduino. Výhody řešení jsou především optimalizace řízení po softwarové stránce, záměna ovládání přes Bluetooth technologií umožňující Wi-Fi komunikaci s větším dosahem a samozřejmě značná úspora místa i hmotnosti.

Byly vytvořeny dvě aplikace v programovacím jazyce C# s podporou platformy .NET Framework. Jedna z nich běží na počítači operátora a jedná se o řídicí program komunikující skrze protokol TCP/IP s druhou aplikací, která je spuštěna na Netduinu. Pro demonstraci funkčnosti komunikace je mikrokontrolér schopen ovládat za pomoci sériového rozhraní na základně povelů operátora subsystém podvozku, tedy motory pásů, světla a blinkry.

Možnosti, jakými lze konečnou variantu práce v rámci robotu Ares dále rozšířit, jsou shledány zejména v oblasti doplnění ovládání ramene robotu a s tím související komunikací se zařízeními EPOS. Dále pak lze uvažovat o koupi nástavby s CAN sběrnici pro Netduino a propojení s první EPOSkou.

Seznam použitých pramenů

- [1] Ares - mobilní robot katedry. *Katedra robotiky: Fakulta strojní, VŠB-TU Ostrava* [online]. ©2016 [cit. 2015-10-03]. Dostupné z: <http://robot.vsb.cz/mobilni-roboty/ares/>
- [2] STUDÉNKA, Marek. Návrh modulární konstrukce pásového podvozku s proměnnou kinematikou rámu. Ostrava, 2010. Diplomová práce. VŠB-TU Ostrava. Vedoucí práce Ladislav Kárník.
- [3] Hercules - mobilní robot katedry. *Katedra robotiky: Fakulta strojní, VŠB-TU Ostrava* [online]. ©2016 [cit. 2015-10-03]. Dostupné z: <http://robot.vsb.cz/mobilni-roboty/hercules/>
- [4] ATAS ELEKTROMOTORY NÁCHOD A.S. *P2X_en.pdf: DC motors PM P2X...* [online]. 2011 [cit. 2015-10-16]. Dostupné z: http://www.atas.cz/files/P2X_en.pdf
- [5] WOLFGANG GRAF - ELEKTROMOTOREN. *P2xr492_d: Stejnoseměrné elektromotory s permanentními magnety* [online]. [cit. 2015-10-11]. Dostupné z: http://www.grafmotoren.eu/technik/p2xr492_d.pdf
- [6] VARVEL SPA. *C_rs-rt_ed01-2011_rev06_gb_271112: RS/RT worm gear boxes* [online]. 2011 [cit. 2015-10-11]. Dostupné z: <http://www.epogearmot.cz/stahovani/category/1-nekove-pevodovky-typ-mrs-mrt-.html?download=1%3Akatalog-rs-rt>
- [7] PG DRIVES TECHNOLOGY. *I-drive overview* [CD]. 2006 [cit. 2015-10-07].
- [8] PG DRIVES TECHNOLOGY. *I-drive manual* [CD]. 2006 [cit. 2015-10-07].
- [9] NOVA-GM45A - IEI Management. *IEI Smart City Solutions* [online]. 2010 [cit. 2015-10-11]. Dostupné z: <http://www.ieiworld.com/publication/news/content.aspx?id=0A075581140874762561>
- [10] VICOR CORPORATION. *Ds_24vin-micro-family: DC-DC Converter Module*. 2015 [cit. 2015-10-23]. Dostupné také z: http://cdn.vicorpower.com/documents/datasheets/ds_24vin-micro-family.pdf
- [11] MAXON MOTOR. *300583-Hardware-Reference-En.pdf: EPOS Positioning Controller* [online]. 2008 [cit. 2015-11-8]. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8803613802526/300583-Hardware-Reference-En.pdf
- [12] MAXON MOTOR: *EPOS2 50/5, Digital positioning controller*. Maxon motor [online]. 2015 [cit. 2015-11-13]. Dostupné z: <http://www.maxonmotor.com/maxon/view/product/control/Positionierung/347717>

- [13] MAXON MOTOR. 15-383-384-385-387-EN.pdf: EPOS2 Positioning Controllers [online]. 2015 [cit. 2015-11-8]. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8816815177758/15-383-384-385-387-EN.pdf
- [14] MAXON MOTOR. 15-267-EN.pdf: EC 90 fl at Ø90 mm, brushless, 90 Watt. 2015. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8816807247902/15-267-EN.pdf
- [15] MAXON MOTOR. PR-EC-90-flat-MILE-en.pdf: EC 90 flat with MILE Encoder. 2012. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8800124534814/PR-EC-90-flat-MILE-en.pdf
- [16] MAXON MOTOR. 15-344-EN.pdf: Encoder MILE 512–6400 CPT. 2015. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8816812490782/15-344-EN.pdf
- [17] MAXON MOTOR. 15-140-EN.pdf: RE 35 Ø35 mm. 2015. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8816798859294/15-140-EN.pdf
- [18] MAXON MOTOR. 15-307-308-EN.pdf: Planetary Gearhead GP 32 C. 2015. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8816808984606/15-307-308-EN.pdf
- [19] MAXON MOTOR. 15-364-365-366-367-EN-Juni.pdf: Encoder HEDL 5540. 2015. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8817170645022/15-364-365-366-367-EN-Juni.pdf
- [20] MAXON MOTOR. 13-346-347-348-EN.pdf: Brake AB 20. 2013. Dostupné také z: http://www.maxonmotor.com/medias/sys_master/root/8807087865886/13-346-347-348-EN.pdf
- [21] Maxon motor e-shop. Maxon motor [online]. 2015 [cit. 2015-11-29]. Dostupné z: http://www.maxonmotor.com/maxon/view/service_search?query=341514
- [22] HARMONIC DRIVE AG. 2014_12_ED_1019642_CSG_2UH.pdf: Engineering Data CSG-2UH Units. 2014. Dostupné také z: http://harmonicdrive.de/produkte/media/catalog/category/2014_12_ED_1019642_CSG_2UH.pdf
- [23] Netduino plus 2: Technical specifications. Netduino [online]. 2014 [cit. 2016-04-13]. Dostupné z: <http://www.netduino.com/netduinoplus2/specs.htm>

- [24] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi: uživatelská příručka*. 1. vyd. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
- [25] Raspberry Pi 3 model B. *Raspberry Pi Foundation* [online]. UK, 2016 [cit. 2016-04-17]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [26] BEHROUZ A. FOROUZAN a WITH SOPHIA CHUNG FEGAN. TCP/IP protocol suite. International ed. Boston, Mass: McCraw Hill, 2007. ISBN 978-007-1260-664.
- [27] LwIP 1.3.0 [online]. 2008 [cit. 2016-04-18]. Dostupné z: <http://www.nongnu.org/lwip/>
- [28] REID, Fiach. Network programming in .NET: with C# and Visual Basic .NET. Boston: Elsevier/Digital Press, c2004. ISBN 1555583156.
- [29] STM32F4 Discovery. STMicroelectronics [online]. 2016 [cit. 2016-04-21]. Dostupné z: http://www2.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-discovery-kits/stm32f4discovery.html
- [30] GOFTON, Peter W. *Sériová komunikace*. Vyd. 1. Překlad Bohumil Kvapil. Praha: Grada, 1994. ISBN 80-716-9131-3.
- [31] BAI, Ying. The windows serial port programming handbook. 1. vydání. Boca Raton, Fla.: Auerbach Publications, c2005. ISBN 08-493-2213-8.

Seznam příloh

Příloha A – Schéma zapojení elektronických součástí robotu Ares [výkres, CD]

Příloha B – Získané výrobní listy jednotlivých elektronických prvků robotu Ares [CD]

Příloha C – Demonstrační úloha komunikace v počítačové síti [CD]

Příloha D – Testovací úloha finální podoby řídicích aplikací pro robot Ares [CD]

Děkuji panu Ing. Jánů Babjakovi, Ph.D. za poskytnutí cenných rad a informací
při vedení bakalářské práce.